Writeup

• For this assignment, your load balancer distributed load based on number of requests the servers had already serviced, and how many failed. A more realistic implementation would consider performance attributes from the machine running the server. Why was this not used for this assignment?

The more realistic implementation was not used because our asgn2 httpserver had no way of returning that data to the load balancer. We only implemented the simple healthcheck, so that is what we used.

• This load balancer does no processing of the client request. What improvements could you achieve by removing that restriction? What would be cost of those improvements?

One possible improvement would be to parse the content length of requests and factor that into the load balancing. Because large files would presumably take longer, we would not send back to back large files to the same server. The cost of these improvements however would be a significant increase in complexity for our load balancer.

• Using your httpserver from Assignment 2, do the following:
– Place eight different large files in a single directory. The files should be around 400 MiB long.
– Start two instances of your httpserver in that directory with four worker threads for each.
– Start your loadbalancer with the port numbers of the two running servers and maximum of eight connections.
– Start eight separate instances of the client at the same time, one GETting each of the files and measure (using time(1)) how long it takes to get the files. The best way to do this is to write a simple shell script (command file) that starts eight copies of the client program in the background, by using & at the end.

Total time: 12.265sec

• Repeat the same experiment, but substitute one of the instances of httpserver for nc, which will not respond to requests but will accept connections. Is there any difference in performance? What do you observe?

Total time: 22.342sec

The difference in performance is roughly 2x slower due to half of the servers being down. This makes it similar to just sending 8 GETs to a single httpserver, just with a little more overhead because of the loadbalancer.